IN THE CLAIMS

Please amend the claims as follows:

1. (Currently Amended) A method for brokering state information exchanged between computers using at least one protocol above a transport layer, the method comprising the steps of:

receiving at a transparent proxy a request from a client requesting a resource of an origin server, wherein the transparent proxy is unknown to the client;

redirecting the client request from the transparent proxy to a policy module;

obtaining at the transparent proxy policy enforcement data, ~~wherein~~ the policy enforcement data is received from the policy module and ~~wherein~~ the policy module and the transparent proxy reside within a same environment;

generating at the transparent proxy a policy state token in response to the policy enforcement data; and

transmitting the policy state token from the transparent proxy to the client, ~~wherein~~ the policy state token is used as an authentication of the client to the transparent proxy for subsequent interactions between the client and the transparent proxy, and the policy state token is represented as a transparent proxy cookie that maintains a relationship among the client, transparent proxy, and the origin server, the transparent proxy cookie includes an indication to the transparent proxy that the client has been authorized by the policy module to use the transparent proxy to access the resource of the origin server, the indication represented as a key whose checksum is verified by the transparent proxy.

2. (Previously Presented) The method of claim 1, further comprising the step of receiving at the transparent proxy a renewed request for the origin server resource, the renewed request containing the policy state token.

3. (Previously Presented) The method of claim 2, wherein the renewed request contains the policy state token in the transparent proxy cookie in a header sent from the client to the transparent proxy.

4. (Original) The method of claim 2, further comprising the step of forwarding to the origin server a portion of the renewed request, the forwarded portion omitting the policy state token.

5. (Previously Presented) The method of claim 4, further comprising the step of receiving at the transparent proxy a reply from the origin server, the reply containing an origin state token for use by the proxy in its subsequent communications with the origin server.

6. (Previously Presented) The method of claim 4, further comprising the steps at the transparent proxy of forwarding to the client at least a portion of a communication from the origin server, and forwarding to the origin server at least a portion of a communication from the client.

7. (Original) The method of claim 1, wherein HTTP is a protocol used during at least one of the receiving and transmitting steps.

8. (Original) The method of claim 1, wherein HTTPS is a protocol used during at least one of the receiving and transmitting steps.

9. (Previously Presented) The method of claim 1, wherein the method further comprises utilizing Novell Directory Services software to provide authentication information about the client, and the transparent policy enforcement data obtained by the transparent proxy depends on the authentication information thus provided.

10. (Previously Presented) The method of claim 1, wherein the method further comprises utilizing Lightweight Directory Access Protocol software to provide authentication

information about the client, and the policy enforcement data obtained by the transparent proxy depends on the authentication information thus provided.

11. (Previously Presented) The method of claim 1, wherein the method further comprises utilizing Secure Sockets Layer software to provide authentication information about the client, and the policy enforcement data obtained by the transparent proxy depends on the authentication information thus provided.

12. (Original) The method of claim 1, wherein the obtaining step extracts policy enforcement data from a redirection address field.

13. (Previously Presented) The method of claim 1, wherein the transmitting step transmits the policy state token in the transparent proxy cookie in a header sent from the transparent proxy to the client.

14. (Currently Amended) A transparent proxy server comprising:

a memory configured at least in part by a transparent proxy process;

a processor for running the transparent proxy process;

at least one link for networked communication between the transparent proxy process, on the one hand, and a client computer and an origin server, on the other hand; and

a policy module identifier which identifies a policy module that grants or denies authorization of proxy services to the client computer by acquiring policy enforcement data and attempting to authenticate the client computer to the transparent proxy process in response to the policy enforcement data, and ~~wherein~~ the client computer directs a request for a resource to an origin server and the request is intercepted by the transparent proxy process, which is unknown to the client computer, and used to determine the policy module identifier which identifies the policy module, and ~~wherein~~ the policy module authenticates the client computer to the transparent proxy process for subsequent interactions between the client computer and the transparent proxy process, and ~~wherein~~

the policy module processes within a same environment as the transparent process, and the transparent proxy creates a transparent proxy cookie for the relationship among the client, the origin server, and the transparent proxy, and the transparent proxy manages ~~managed~~ the transparent proxy cookie on the client, the transparent proxy cookie includes an indication to the transparent proxy that the client has been authorized by the policy module to use the transparent proxy to access the resource of the origin server, the indication represented as a digital signature that is verified by the transparent proxy.

15. (Original) The transparent proxy server of claim 14, in combination with the policy module.

16. (Original) The transparent proxy server of claim 15, wherein the policy module and the transparent proxy process are running on the same computer.

17. (Original) The transparent proxy server of claim 14, in combination with the client computer and at least one other client computer, each client computer linked for networked communication with the transparent proxy process.

18. (Previously Presented) The transparent proxy server of claim 14, wherein the transparent proxy server provides authorized proxy service transparently to both the client computer and the origin server by steps which comprise receiving the request from the client for the resource of the origin server, sending the client computer an authorization by the policy module for the client computer to use a transparent proxy service, accepting the authorization from the client computer with a renewed client request for the origin server resource, forwarding the renewed client request to the origin server without forwarding the authorization but with an indication to the origin server that the transparent proxy server is the source of the forwarded request, and then transparently forwarding the requested resource from the origin server to the client computer.

19. (Previously Presented) The transparent proxy server of claim 18, wherein the transparent proxy server sends the client computer the authorization by sending the client computer the transparent proxy cookie for use in subsequent communications from the client computer.

20. (Original) The transparent proxy server of claim 14, in combination with at least one additional transparent proxy server which also has a memory configured at least in part by a transparent proxy process, a processor for running the transparent proxy process, a link, and a policy module identifier.

21. (Previously Presented) The combined transparent proxy servers of claim 20, wherein one transparent proxy server forwards other client requests to the other transparent proxy server.

22. (Original) The combined transparent proxy servers of claim 20, wherein one transparent proxy server takes over the handling of client requests in place of the other transparent proxy server.

23-31. (Cancelled)